FLEX - RPC SERVICES

Flex provides RPC services to provide server side data to client side. Flex provides a fair amount of control on to server side data.

- Using Flex RPC services, we can define user actions to be executed on server side.
- Flex RPC Sservices can be integrated with any server side technologies.
- One of Flex RPC Service provide inbuilt support for compressed binary data to be transferred over the wire and is pretty fast.

Flex provides following three types of RPC Services

RPC Service	Description
HttpService	<mx:HTTPService> tag is used to represent an HTTPService object in an MXML file. When you make a call to HTTPService object's send() method, it makes an HTTP request to the specified URL, and an HTTP response is returned.You can also use the HTTP HEAD, OPTIONS, TRACE, and DELETE methods.
WebService	The <mx:WebService> tag is used to get access to the operations of SOAP-compliant web services.
RemoteObject	The <mx:RemoteObject> tag is used to represent an HTTPService object in an MXML file. This tag gives you access to the methods of Java objects using Action Message Format (AMF) encoding.

We're going to discuss HTTP Service in detail. We'll use an XML source file placed at server and access it at client side via HTTP Service

Items.xml

```
<items>
   <item name="Book" description="History of France"></item>
   <item name="Pen" description="Parker Pen"></item>
   <item name="Pencil" description="Stationary"></item>
<items>
```

HTTPService Declaration

Now declare a HTTPService and pass it url of the above file

```
<fx:Declarations>
   <mx:HTTPService
   url="http://www.tutorialspoint.com/flex/Items.xml" />
</fx:Declarations>
```

RPC Call

Make a call to itemRequest.send() method and bind values from lastResult object of itemRequest webservice to Flex UI component.

```
...
itemRequest.send();
```

```
...
<mx:DataGrid
   dataProvider="{itemRequest.lastResult.items.item}">
   <mx:columns>
      <mx:DataGridColumn headerText="Name" dataField="name"/>
      <mx:DataGridColumn headerText="Description" dataField="description"/>
   </mx:columns>
</mx:DataGrid>
```

RPC Service Call Example

Now Let us follow the following steps to test RPC services in a Flex application:

Step	Description
1	Create a project with a name *HelloWorld* under a package *com.tutorialspoint.client* as explained in the *Flex - Create Application* chapter.
2	Modify *HelloWorld.mxml* as explained below. Keep rest of the files unchanged.
3	Compile and run the application to make sure business logic is working as per the requirements.

Following is the content of the modified mxml file **src/com.tutorialspoint/HelloWorld.mxml**.

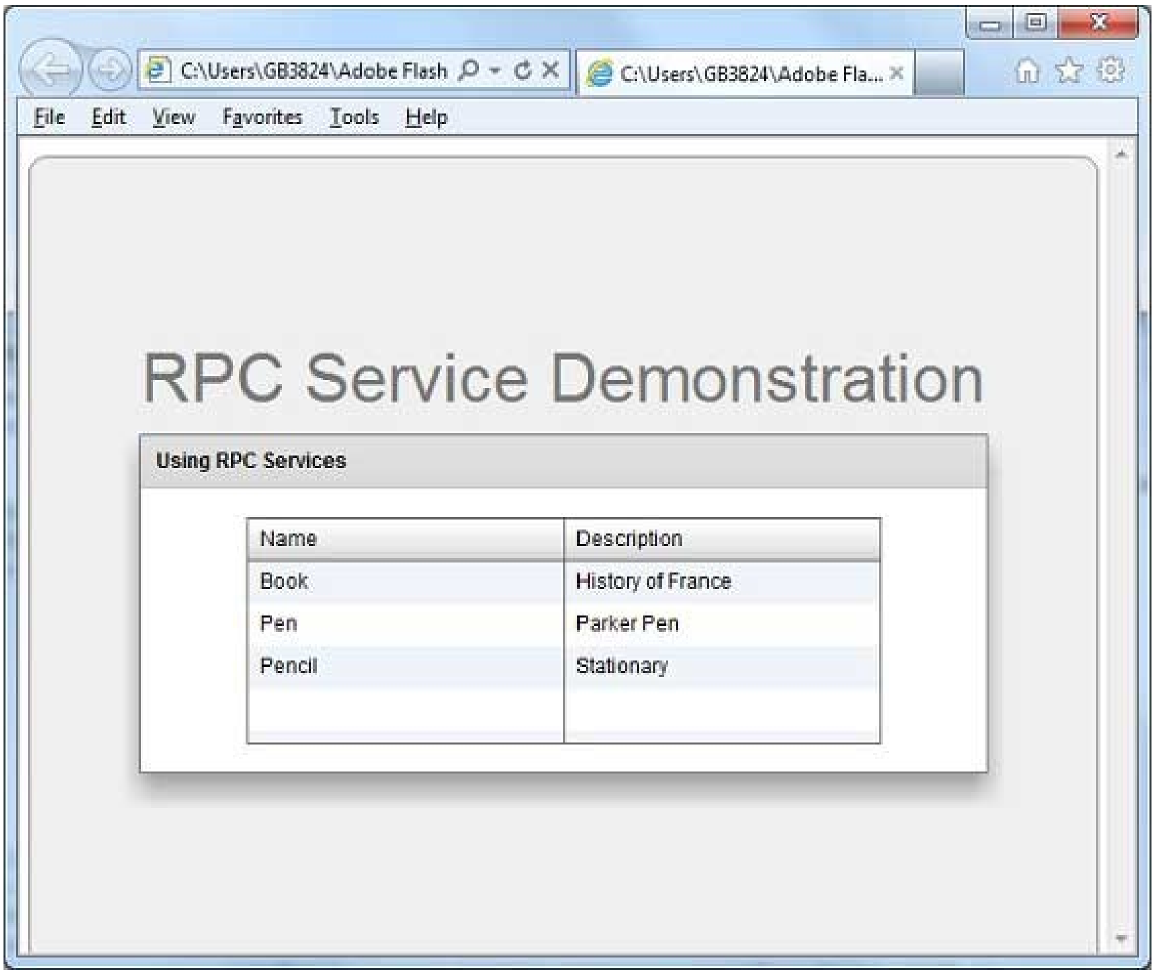
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:mx="library://ns.adobe.com/flex/mx"
   minWidth="500" minHeight="500" creationComplete="init(event)">
   <fx:Style source="/com/tutorialspoint/client/Style.css"/>
   <fx:Script>
      <![CDATA[
         import mx.events.FlexEvent;
         import mx.rpc.events.FaultEvent;
         import mx.rpc.events.ResultEvent;

         protected function init(event:FlexEvent):void
         {
            itemRequest.send();
         }

      ]]>
   </fx:Script>
   <fx:Declarations>
      <mx:HTTPService
      url="http://www.tutorialspoint.com/flex/Items.xml" />
   </fx:Declarations>
   <s:BorderContainer width="630" height="480"
      styleName="container">
      <s:VGroup width="100%" height="100%" gap="10"
         horizontalAlign="center" verticalAlign="middle">
         <s:Label
            fontSize="40" color="0x777777" styleName="heading"/>
         <s:Panel
            width="500" height="200" >
            <s:layout>
               <s:VerticalLayout  gap="10"
                  verticalAlign="middle" horizontalAlign="center"/>
            </s:layout>
            <mx:DataGrid
               dataProvider="{itemRequest.lastResult.items.item}">
               <mx:columns>
                  <mx:DataGridColumn headerText="Name"
                     dataField="name"/>
                  <mx:DataGridColumn headerText="Description"
```

```
                    dataField="description"/>
                </mx:columns>
            </mx:DataGrid>
        </s:Panel>
    </s:VGroup>
</s:BorderContainer>
</s:Application>
```

Once you are ready with all the changes done, let us compile and run the application in normal mode as we did in Flex - Create Application chapter. If everything is fine with your application, this will produce following result: [Try it online]



RPC Service Demonstration

Using RPC Services

Name	Description
Book	History of France
Pen	Parker Pen
Pencil	Stationary